CHIRIPA Documentation

Release 1.0

Javier Ramos

*Macromolecular Physics Department, Instituto de Estructura de la Materia, CSIC, Serrano 113bis, 28006, Madrid (Spain)*

22/10/2019

**Table of Contents**

[1. Theoretical background 3](#_Toc55991604)

[1. Outline 4](#_Toc55991605)

[2. Theoretical details 4](#_Toc55991606)

[2.1. Definition of the repeat units. 4](#_Toc55991607)

[2.2. Aa 5](#_Toc55991608)

[2.3. Sss 5](#_Toc55991609)

[2.4. XXXXX 5](#_Toc55991610)

[1. Example 1: Hexane/Nitrobenzene (in steps) 17](#_Toc55991611)

[1.1. Introduction 18](#_Toc55991612)

[1.2. Preparation of the structures 18](#_Toc55991613)

[1.3. Coordination number calculations (Zij)using Vander Waals radii 20](#_Toc55991614)

[1.4. Coordination numbers calculation (Zij)using Okuwaki revaluated distances. 23](#_Toc55991615)

[1.5. Anisotropy calculation (Sij)using Vander Waals radii 24](#_Toc55991616)

[2. Example 2: Hexane/Nitrobenzene (automatic) 26](#_Toc55991617)

[2.1. Introduction 27](#_Toc55991618)

[2.2. Preparing the input 27](#_Toc55991619)

[3. Classes in Chiripa 30](#_Toc55991620)

[1. Segment class 31](#_Toc55991621)

[1.1. ddddd 31](#_Toc55991622)

[2. References 35](#_Toc55991623)

|  |
| --- |
|  |

# Theoretical background

|  |
| --- |
|  |

## Outline

CHIRIPA is a Python framework that allows to calculate the interaction parameter (**χij**) by determining the interactions between two monomers at a molecular level using the methods proposed by Blanco-Fan 1,2 and Okuwaki et al.3,4.

CHIRIPA stands for CHI inteRactIon PArameter. Chiripa is a Spanish word that means “by luck” making reference to the use of random numbers in the proposed algorithms.

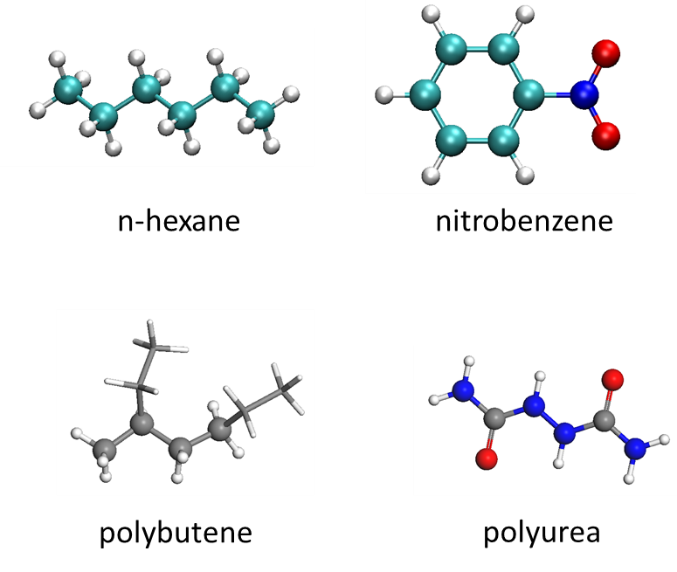
## Theoretical details

In this section, we briefly summarize the theoretical and some aspects of the implemented algorithms in CHIRIPA.

In few words, Fan et al.2 claimed that the interaction parameter (**χij**) of two polymer repeat units (or segments) can be calculated from the pairwise contact energies of an ensemble of pairs of the segments. In order to solve this problem, a number of steps needs to be performed.

### Definition of the repeat units.

We need to construct proper structures for the molecules of interest. This should be the whole molecule in the case of solvent or small molecules or a group of repeat units in the case of polymers. **Figure 1** shows some examples.



**Figure 1.** Examples of molecular structures of segments to be used in Chiripa. In the case of polymers a dimer can be used.

A pdb, xyz or gro file can be used in CHIRIPA. The segment should be optimized using DFT methods before to be used.

An object Segment is used to represent the molecules.

### Aa

### Sss

### XXXXX

The available keywords used in the input dictionary defining a calculation using CHIRIPA library are the following:

* [names](#key_names)
* [filecoords](#key_filecords)
* [filetop](#key_top)

#### names

Description

A list of strings containing the names of the molecules to be used. This list contains the names used along the program to identify the segments (or molecules) involved in the calculation. The first name corresponds to the molecule number 1 and the second to the number 2 in the output files. The order of [names](#key_names), [filecords](#key_filecords) and [filetops](#key_top) must be consistent, please check the info in the output file (output.log)

Type

List of strings of length 2

Allowed values

Any string

Default or Required

Required

Examples

[“n-hexane”, “nitrobenzene”]

filecords

Description

A list of strings containing the names of coordinate files. This list contains the filenames used to read the coordinates of the segments. The first value corresponds to the molecule number 1 and the second to the number 2 in the output files. The order of the filenames in this list must be consistent with the names in the [names](#key_names) keyword. Coordinates formats accepted are pdb, xyz and gro.

TODO: The xyz and gro format must be tested in the Chi\_Universe

Type

List of strings of length 2

Allowed values

Any string

Default or Required

Required

Examples

[“n-hexane.pdb”, “nitrobenzene.pdb”]

Note

Inputdict = {‘names’: [“n-hexane.pdb”, “nitrobenzene.pdb”],

‘filecoords’: [“n-hexane.pdb”, “nitrobenzene.pdb”]}

filetop

Description

A list of strings containing the names of topology files. This list contains the filenames used to read or guess the topology of the segments. The first value corresponds to the molecule number 1 and the second to the number 2 in the output files. The order of the filenames in this list must be consistent with the names in the [names](#key_names) keyword. Topology formats accepted are pdb, xyz and gro.

For pdb files if the “CONECT” section exists, the topology is built from this section, otherwise the topology is guessed based on the distance of the atoms using the formula [Zhang, Q.; Zhang, W.; Li, Y.; Wang, J.; Zhang, L.; Hou, T., A rule-based algorithm for automatic bond type perception. *Journal of Cheminformatics* **2012,** *4* (1), 26]:

, where ri and rj are the covalent radii of the atoms (supplement information in [Zhang, Q.; Zhang, W.; Li, Y.; Wang, J.; Zhang, L.; Hou, T., A rule-based algorithm for automatic bond type perception. *Journal of Cheminformatics* **2012,** *4* (1), 26].

For xyz and gro files the topology is always guessed.

TODO: The xyz and gro format must be tested in the Chi\_Universe. The psf and top topology should be included.

Type

List of strings of length 2

Allowed values

Any string

Default or Required

Required

Examples

[“n-hexane.pdb”, “nitrobenzene.pdb”]

Note

Inputdict = {‘names’: [“n-hexane.pdb”, “nitrobenzene.pdb”],

‘filecoords’: [“n-hexane.pdb”, “nitrobenzene.pdb”],

‘filetop’: [“n-hexane.pdb”, “nitrobenzene.pdb”],}

coordination\_numbers\_Z

Description

A flag indicating if the calculation of the coordination number (Z) is done. The results are written to the file Z\_results.log. If the file exists in the current folder, the calculation is not performed although this flag is True. This file should need to be moved, renamed or deleted in the current folder to make the calculation.

By default the parameters used in the Z calculation are:

* Number of samples to calculate coordination number (Z) = 20
* Number of trials to put a Segment = 1000
* Number of processes available = 2
* Number of tasks in parallel = 4

See section …. for details

Type

Boolean

Allowed values

True/False

Default or Required

False

Z\_parameters

Description

A dictionary containing the parameters to perform the coordination number (Z). The allowed keywords are:

* “Z\_samples“: Number of replicas for the Z calculation for a given pair of segments. The final result is the mean value of all replicas.
* “Z\_puttrialmonomers”: Maximum number of trials to put the current monomer around the central one. The algorithm ends when the number of trials exceeds this number.
* “Z\_debug”: This flag is used to write the generated conformations for each sample in each pair. If True four folders are generated (aZ11\_coordination, aZ12\_coordination, aZ21\_coordination and aZ22\_coordination) containing the conformations.
* “Z\_nonbonded”: Types of van der Waals (VdW) radii used in the evaluation of the nonbonding contacts among atoms. So far, two values are allowed:
  + “thrular”: The VdW values published in Mantina et al. (Table 12)[ Mantina, M.; Chamberlin, A. C.; Valero, R.; Cramer, C. J.; Truhlar, D. G., Consistent van der Waals Radii for the Whole Main Group. *The Journal of Physical Chemistry A* **2009,** *113* (19), 5806-5812] are used.
  + “okuwaki\_correction”. The reevaluated distances calculated by Okuwaki et al. (Table 2) are used.

A flag indicating if the calculation of the coordination number (Z) is done. The results are written to the file Z\_results.log. If this file already exists in the current folder, the calculation is not performed although this flag be True. This file needs to be renamed or deleted to make the calculation.

By default the parameters used in the Z calculation are:

* Number of samples to calculate coordination number (Z) = 20
* Number of trials to put a Segment = 1000
* Number of processes available = 2
* Number of tasks in parallel = 4

See section …. for details

Type

Dictionary

Allowed values

“Z\_samples“ : An integer

“Z\_puttrialmonomers“ : An integer

“Z\_debug“ : A boolean

“Z\_nonbonded“ : “truhlar” or “okuwaki\_correction”

Default or Required

“Z\_samples“ : 20

“Z\_puttrialmonomers“ : 1000

“Z\_debug“ : False

“Z\_nonbonded“ : “truhlar”

Examples

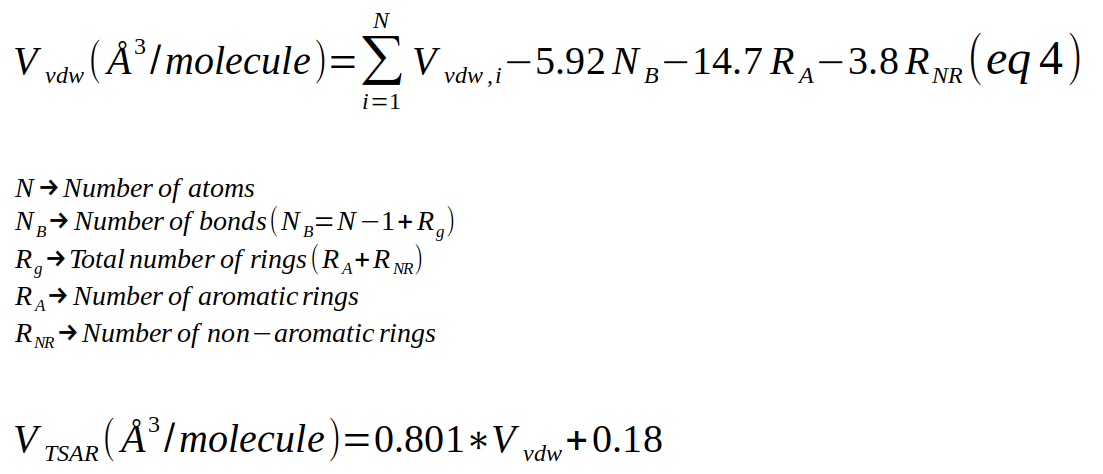
1. inputdict = { 'names'                  : ["n-hexane", "nitrobenzene"],
2. 'filecoords'             : ["./n-hexane.pdb", "./nitrobenzene.pdb"],
3. 'filetop'                : ["./n-hexane.pdb", "./nitrobenzene.pdb"],
4. 'coordination\_numbers\_Z' : True,
5. 'Z\_parameters'           : {'Z\_samples'         :  100,
6. 'Z\_puttrialmonomers': 3000,
7. 'Z\_debug'           : False,
8. 'Z\_nonbonded'       : 'truhlar'},

calculate\_volume

Description

Calculation of the van der Waals volume using the method reported by Zhao et al. [Zhao, Y. H.; Abraham, M. H.; Zissimos, A. M., Fast Calculation of van der Waals Volume as a Sum of Atomic and Bond Contributions and Its Application to Drug Compounds. *The Journal of Organic Chemistry* **2003,** *68* (19), 7368-7373 ]

The VdW radii are taken from Bondi A et al. [Bondi, A., van der Waals Volumes and Radii. *The Journal of Physical Chemistry* **1964,** *68* (3), 441-451.] except the value for H, which is taken from Rowland RS et al.[ Rowland, R. S.; Taylor, R., Intermolecular Nonbonded Contact Distances in Organic Crystal Structures: Comparison with Distances Expected from van der Waals Radii. *The Journal of Physical Chemistry* **1996,** *100* (18), 7384-7391]. Radii that are not available in either of these publications have a value of 2.0 Å. The radii for Ions (Na, K, Cl, Ca, Mg, and Cs) are based on the CHARMM27 force field.

The formula (4) of the Zhao’s article will be used in this function:

Type

Boolean

Allowed values

True/False

Default or Required

False

interaction\_energy

Description

A flag indicating if the interaction energy between segments is performed. The interaction energies are calculated following the energy\_parameters keyword.

Type

Boolean

Allowed values

True/False

Default or Required

False

calculate\_volume

Description

A dictionary containing the parameters to perform the coordination number (Z). The allowed keywords are:

Type

Dictionary

Allowed values

“Z\_samples“ : An integer

“Z\_puttrialmonomers“ : An integer

“Z\_debug“ : A boolean

“Z\_nonbonded“ : “truhlar” or “okuwaki\_correction”

Default or Required

“Z\_samples“ : 20

“Z\_puttrialmonomers“ : 1000

“Z\_debug“ : False

“Z\_nonbonded“ : “truhlar”

Examples

1. inputdict = { 'names'                  : ["n-hexane", "nitrobenzene"],
2. 'filecoords'             : ["./n-hexane.pdb", "./nitrobenzene.pdb"],
3. 'filetop'                : ["./n-hexane.pdb", "./nitrobenzene.pdb"],
4. 'coordination\_numbers\_Z' : True,
5. 'Z\_parameters'           : {'Z\_samples'         :  100,
6. 'Z\_puttrialmonomers': 3000,
7. 'Z\_debug'           : False,
8. 'Z\_nonbonded'       : 'truhlar'},

|  |
| --- |
|  |

# Example 1: Hexane/Nitrobenzene (in steps)

|  |
| --- |
|  |

## Introduction

The n-hexane **(1)** / nitrobenzene **(2)** binary system represents a standard example to calculate the interaction parameter (**χ**). This system has been already studied by Fan et al.[ Fan, C. F.; Olafson, B. D.; Blanco, M.; Hsu, S. L., Application of molecular simulation to derive phase diagrams of binary mixtures. *Macromolecules* **1992,** *25* (14), 3667-3676] {Fan, 1992 #2}using the method of neighbor contacts between segments described in its paper. Furthermore, this system was used by Okuwaki et al. to assess the reliability of its method based on neighbor contacts and fragment molecular orbital (FMO) calculations. We use this standard binary system to test CHIRIPA code implementation. Therefore, the results from CHIRIPA will be compared with those reported by Fan et al. and Okuwaki et al.

## Preparation of the structures

Both n-hexane and nitrobenzene are sketched in Materials Studio 2019 (MS2019) and optimized using PW91/DNP method using DMol3. Figure 1 shows the optimized molecular structures. The pdb files are obtained by exporting the structure directly from MS2019.

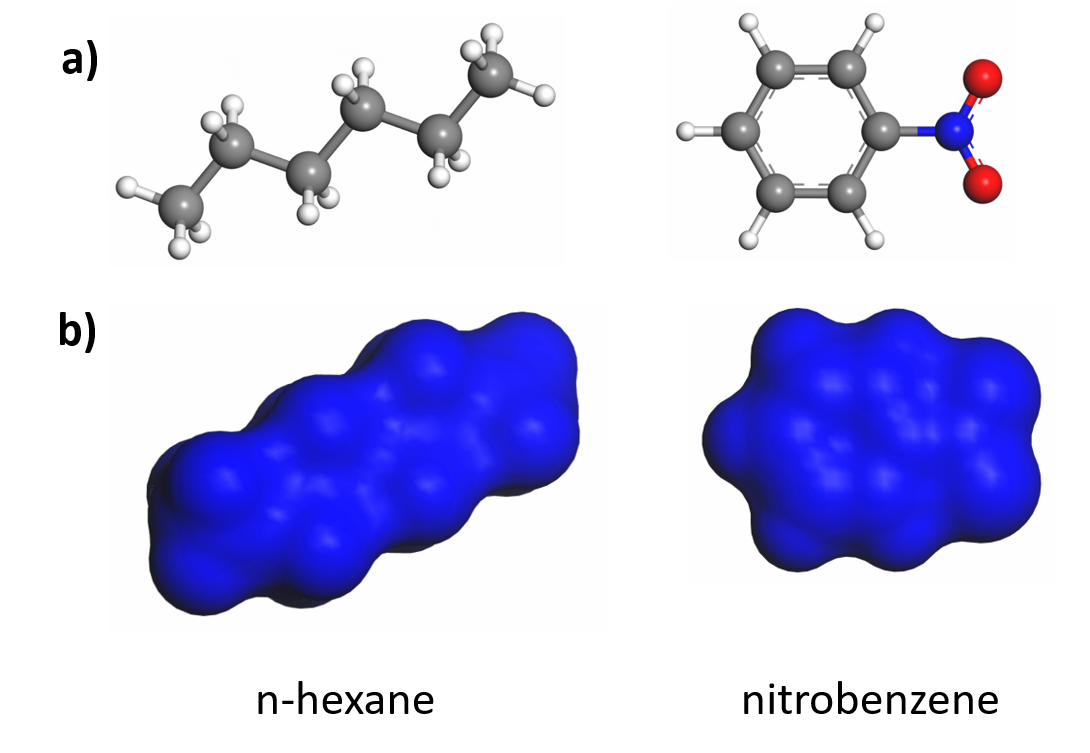


Figure 2: Molecular structures used for this example. a) DFT-optimized and b) Connolly surface for both structures. The Connolly structures were generated with a grid interval of 0.25 Å, VdW scale factor of 1.0 Å and Connolly radius of 0 Å. The van der Waals volume is equivalent to calculate the Connolly surface with a Connolly radius of 0 Å.

The Connolly surface for C6H14 and C6H4NO2 is shown in Figure 1. The Van der Waals volume is 111.7 and 105.1 Å3/molecule for n-hexane and nitrobenzene, respectively. On the other hand, the accessible surface volume based on a probe of 1.4 Å are 436 and 394 Å3/molecule, respectively. Alternatively, the van der Waals volume can be estimated by CHIRIPA (**Code Box 1**) using the method proposed by Zhao et al. [Zhao] giving 112.4 and 107.1 Å3/molecule, respectively. The VdW volumes reported by Fan et al. [Fan] are 180 and 147 Å3/molecule for n-hexane and nitrobenzene. These volumes are 61% and 40% greater than those calculated with MS2019 and Zhao method.

Irrespective of the method used to calculated the VvdW, volumes for both molecules are similar, which is a requirement of the method to calculate the **χ** parameter.

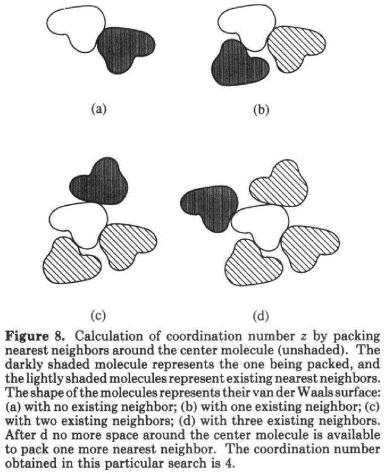


Figure 3: Z calculation in the Fan et al. article.

**Code Box 1:** Get the VdW volumes of two segments.

1. **from** chipar.Simulator **import** Segment
3. # Load n-hexane
4. s1 = Segment(filecoord="./n-hexane.pdb", filetop="./n-hexane.pdb")
5. v\_vdw, v\_tsar = s1.calc\_vdw\_volume\_VABC()
7. **print**("VdW volume for n-hexane = {:.1f} anstroms^3/molecule".format(v\_vdw))
9. # Load nitrobenzene
10. s2 = Segment(filecoord="./nitrobenzene.pdb", filetop="./nitrobenzene.pdb")
11. v\_vdw, v\_tsar = s2.calc\_vdw\_volume\_VABC()
13. **print**("VdW volume for nitrobenzene = {:.1f} anstroms^3/molecule".format(v\_vdw))

## Coordination number calculations (Zij)using Vander Waals radii

The **Code Box 2** contains the script to calculate the coordination number between the n-hexane and nitrobenzene molecules. The first step is generating the segments for each monomer using the class Segment. The constructor of the class needs both the coordinates and topology files, in this example “./n-hexane.pdb” and “./nitrobencene.pdb”. After, the four **Zij** are calculated by calling the calculate\_average\_z\_number function. This function implements the method described by Fan et al. (Figure 2) and Okuwaki et al. (see section 2.1.iii). Two parameters are here important, samples\_Z (30) and trialmonomer (10000). The first parameter is the number of times that the **Zij** is calculated for the pair i-j, the mean value will be the resulting **Zij.** The second parameter is the number of maximum trials to place the **j** segment around the **i** ones containing other **j** segment around (Figure 8 in ref. FAN).

**Code Box 2:** Calculation of Z11, Z12, Z21, and Z22. (example\_01.py)

1. **from** chipar.Segment **import** Segment
2. **from** chipar.Simulator **import** Simulator
3. **from** chipar.CubicBox **import** CubicBox
4. **from** chipar.generate\_pair\_conformations **import** calculate\_average\_z\_number
6. # Parameters
7. name1 = "n-hexane"
8. name2 = "nitrobenzene"
9. samples\_Z = 30
10. trialmonomer = 10000
11. f1 = "./n-hexane.pdb"
12. f2 = "./nitrobenzene.pdb"
14. # Load n-hexane ============================
15. **print**("1. Loading segment {:s}".format(f1))
16. s1 = Segment(filecoord=f1, filetop=f1)
17. v\_vdw, v\_tsar = s1.calc\_vdw\_volume\_VABC()
18. **print**("\tVdW volume for {:s} = {:.1f} anstroms^3/molecule".format(f1, v\_vdw))
20. # Load nitrobenzene ============================
21. **print**("2. Loading segment {:s}".format(f2))
22. s2 = Segment(filecoord=f2, filetop=f2)
23. v\_vdw, v\_tsar = s2.calc\_vdw\_volume\_VABC()
24. **print**("\tVdW volume for {:s} = {:.1f} anstroms^3/molecule".format(f2, v\_vdw))
26. # Calculate the number of coordination Z between pairs
27. Z12, Z12\_std = calculate\_average\_z\_number(s1, s2, samples\_Z= samples\_Z, putTrialsMonomer=trialmonomer,
28. debug=True, idir="./Z12-coordination")
29. Z21, Z21\_std = calculate\_average\_z\_number(s2, s1, samples\_Z= samples\_Z, putTrialsMonomer=trialmonomer,
30. debug=True, idir="./Z21-coordination")
31. Z11, Z11\_std = calculate\_average\_z\_number(s1, s1, samples\_Z= samples\_Z, putTrialsMonomer=trialmonomer,
32. debug=True, idir="./Z11-coordination")
33. Z22, Z22\_std = calculate\_average\_z\_number(s2, s2, samples\_Z= samples\_Z, putTrialsMonomer=trialmonomer,
34. debug=True, idir="./Z22-coordination")

37. **print**("\*\*\*\*\*\*\*\*\*\*\*\* SUMMARY \*\*\*\*\*\*\*\*\*\*\*\*")
38. **print**("Z12 = {:.2f} +- {:.2f}".format(Z12, Z12\_std))
39. **print**("Z21 = {:.2f} +- {:.2f}".format(Z21, Z21\_std))
40. **print**("Z11 = {:.2f} +- {:.2f}".format(Z11, Z11\_std))
41. **print**("Z22 = {:.2f} +- {:.2f}".format(Z22, Z22\_std))
42. **print**("\*\*\*\*\*\*\*\*\*\*\*\* SUMMARY \*\*\*\*\*\*\*\*\*\*\*\*")

The **Zij** number corresponds to the number of **j** molecules neighbors to an **i** central segment. The algorithm places the central molecule (**i**) in a simulation box, then packing nearest neighbors (**j**) around the central molecule until no more space is available (in this implementation when trialmonomer trials are exhausted).

Table 1 (column Zij (VdW)) shows the Zij for the n-hexane (**1**) and nitrobenzene (**2**) pair binary system using the Van der Waals data from Truhlar et al.[Mantina]. The coordination numbers are calculated 30 times using 10000 as maximum trials to put a segment.

Table 1: Coordination numbers for the system n-hexane/nitrobenzene using different approaches

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pair | Zij (VdW)a) | Zij (RevDist)b) | Zij (MS)c) | Zij (Fan)d) | Zij (Okuwaki)e) |
| 1-1 | 11.0±1.03 | 10.7±1.08 |  | n.a | 10.6 |
| 1-2 | 11.0±0.95 | 10.3±1.09 |  | n.a | 10.4 |
| 2-1 | 10.5±1.12 | 10.2±1.15 |  | n.a | 10.7 |
| 2-2 | 10.2±0.97 | 10.6±1.08 |  | n.a | 10.6 |
| Avg | 10.7±1.02 | 10.5±1.1 |  | 11.5 | 10.6 |
| Cpu time(s) | 4970 | 6288 |  | n.a | n.a |

n.a stands for not available. a) Coordination number using the CHIRIPA program and the VdW radius set calculated by Truhlar et al. b) Coordination number using the CHIRIPA program and the non-bonded distances given in Okuwaki et al. c) Coordination number calculated using the module BLENDS presents in MS2019. d) Values given in ref FAN and e) Values given in ref Okuwaki and FAN

## Coordination numbers calculation (Zij)using Okuwaki revaluated distances.

Okuwaki et al. have proposed the correction of the distances using quantum mechanical calculations. These revaluated distances are implemented in CHIRIPA. The **Code Box 3** contains the script to calculate the coordination number between the n-hexane and nitrobenzene molecules using the revaluated distances. Results are collected in Table 1 (column Zij (RevDist)).

**Code Box 3:** Calculation of Z11, Z12, Z21, and Z22. (example\_04.py)

1. **from** chipar.Segment **import** Segment
2. **from** chipar.generate\_pair\_conformations **import** calculate\_average\_z\_number
4. # Parameters
5. name1 = "n-hexane"
6. name2 = "nitrobenzene"
7. samples\_Z = 30
8. trialmonomer = 10000
9. f1 = "./n-hexane.pdb"
10. f1\_type = "./n-hexane\_types.dat"
11. f2 = "./nitrobenzene.pdb"
12. f2\_type = "./nitrobenzene\_types.dat"
14. # Load n-hexane ============================
15. **print**("1. Loading segment {:s}".format(f1))
16. s1 = Segment(filecoord=f1, filetop=f1, filetypeatoms=f1\_type)
17. v\_vdw, v\_tsar = s1.calc\_vdw\_volume\_VABC()
18. **print**("\tVdW volume for {:s} = {:.1f} anstroms^3/molecule".format(f1, v\_vdw))
20. # Load nitrobenzene ============================
21. **print**("2. Loading segment {:s}".format(f2))
22. s2 = Segment(filecoord=f2, filetop=f2, filetypeatoms=f2\_type)
23. v\_vdw, v\_tsar = s2.calc\_vdw\_volume\_VABC()
24. **print**("\tVdW volume for {:s} = {:.1f} anstroms^3/molecule".format(f2, v\_vdw))
26. # Calculate the number of coordination Z between pairs
27. Z12, Z12\_std = calculate\_average\_z\_number(s1, s2, samples\_Z= samples\_Z, putTrialsMonomer=trialmonomer,
28. debug=True, idir="./Z12-coordination",
29. nonbondedEvaluation="okuwaki\_correction")
30. Z21, Z21\_std = calculate\_average\_z\_number(s2, s1, samples\_Z= samples\_Z, putTrialsMonomer=trialmonomer,
31. debug=True, idir="./Z21-coordination",
32. nonbondedEvaluation="okuwaki\_correction")
33. Z11, Z11\_std = calculate\_average\_z\_number(s1, s1, samples\_Z= samples\_Z, putTrialsMonomer=trialmonomer,
34. debug=True, idir="./Z11-coordination",
35. nonbondedEvaluation="okuwaki\_correction")
36. Z22, Z22\_std = calculate\_average\_z\_number(s2, s2, samples\_Z= samples\_Z, putTrialsMonomer=trialmonomer,
37. debug=True, idir="./Z22-coordination",
38. nonbondedEvaluation="okuwaki\_correction")

41. **print**("\*\*\*\*\*\*\*\*\*\*\*\* SUMMARY \*\*\*\*\*\*\*\*\*\*\*\*")
42. **print**("Z12 = {:.2f} +- {:.2f}".format(Z12, Z12\_std))
43. **print**("Z21 = {:.2f} +- {:.2f}".format(Z21, Z21\_std))
44. **print**("Z11 = {:.2f} +- {:.2f}".format(Z11, Z11\_std))
45. **print**("Z22 = {:.2f} +- {:.2f}".format(Z22, Z22\_std))
46. **print**("\*\*\*\*\*\*\*\*\*\*\*\* SUMMARY \*\*\*\*\*\*\*\*\*\*\*\*")

## Anisotropy calculation (Sij)using Vander Waals radii

Another important factor discussed by Okuwaki et al. is the anisotropy correction (Sij). The Flory-Huggins theory assumes spatial isotropic, but depending of the molecular shapes and the specific interactions between the segments this assumption cannot be justified.

(…TODO…..)

|  |
| --- |
|  |

# Example 2: Hexane/Nitrobenzene (automatic)

|  |
| --- |
|  |

## Introduction

In the previous section, a step by step method to calculate all needed parameters has been covered. However, it is convenient to automatize as much as possible all the calculations. Thus, in the current section, we are going to explain the steps to perform the calculation of the interaction parameter for the n-hexane and nitrobenzene in automatic way.

## Preparing the input

The CHIRIPA jobs use a python script to start the calculations. You can control system initialization, server definition, coordination number, energy and anisotropy calculations. An example is giving in **Code Box 4.**

The first is to import the Chi\_Universe class from the Chi\_Universe module (line 1). The keywords to run the calculation are organized as a dictionary commonly called inputdict (line 11-39). The keywords allowed in this dictionary are defined in Table 1.

**Code Box 4:** Automatic calculation (example\_16\_small.py)

1. **from** chiripa.Chi\_Universe **import** Chi\_Universe
3. """ Using Slurm and Gaussian16 in trueno.csic.es"""
5. # Parameters ====================================
6. name1 = "n-hexane"
7. name2 = "nitrobenzene"
8. f1 = "./n-hexane.pdb"
9. f2 = "./nitrobenzene.pdb"
11. inputdict = { 'names'                 : ["n-hexane", "nitrobenzene"],
12. 'filecoords'            : ["./n-hexane.pdb", "./nitrobenzene.pdb"],
13. 'filetop'               : ["./n-hexane.pdb", "./nitrobenzene.pdb"],
14. 'coordination\_numbers\_Z': True,
15. 'Z\_parameters'          : {'Z\_samples'         : 3,
16. 'Z\_puttrialmonomers': 100,
17. 'Z\_debug'           : False,
18. 'Z\_nonbonded'       : 'truhlar'},
19. 'calculate\_volume'      : True,
20. 'interaction\_energy'    : True,
21. 'energy\_parameters'     : {'qm\_engine'            : "NwChem",
22. 'qm\_path\_exe'          : "/home/cfmac/jramos/CODES/NWCHEM/nwchem-6.6\_cpu/bin/LINUX64/nwchem",
23. 'qm\_charge'            : 0,
24. 'qm\_multiplicity'      : 1,
25. 'qm\_scratch\_dir'       : '/scratch-local/',
26. 'qm\_basisset'          : '6-31g\*\*',
27. 'qm\_method'            : 'm06-2x',
28. 'qm\_task'              : 'energy',
29. 'number\_configurations': 5,},
30. 'server'                : {'name'         : "trueno.csic.es",
31. 'queue\_system' : "slurm",
32. 'username'     : 'jramos',
33. 'key\_file'     : '/home/jramos/.ssh/id\_rsa\_chiripa',
34. 'local\_dir'    : '/home/jramos/PycharmProjects/chiripa/examples/EXAMPLE\_16/',
35. 'remote\_dir'   : '/home/cfmac/jramos/CHIRIPA/RUN\_016/',
36. 'ncpus'        : 1,
37. 'nodelist'     : "trueno95"
38. }
39. }
41. # Clean files
42. #Chi\_Universe({}, output\_screen=True, clean="full")
43. c = Chi\_Universe(inputdict, output\_screen=True)

46. **print**("Job Done!!!!")

|  |
| --- |
|  |

# Classes in Chiripa

|  |
| --- |
|  |

## Segment class

### Description

This class is used to model a molecule in CHIRIPA. The UML diagram is shown below:



Methods

Fields

**Figure 4.** UML diagram for the Segment object

### dwdwdwedw

|  |
| --- |
|  |

# References

|  |
| --- |
|  |

(1) Blanco, M. Molecular Silverware. I. General Solutions to Excluded Volume Constrained Problems. *J. Comput. Chem.* **1991**, *12* (2), 237–247. https://doi.org/10.1002/jcc.540120214.

(2) Fan, C. F.; Olafson, B. D.; Blanco, M.; Hsu, S. L. Application of Molecular Simulation To Derive Phase Diagrams of Binary Mixtures. *Macromolecules* **1992**, *25* (14), 3667–3676. https://doi.org/10.1021/ma00040a010.

(3) Okuwaki, K.; Mochizuki, Y.; Doi, H.; Ozawa, T. Fragment Molecular Orbital Based Parametrization Procedure for Mesoscopic Structure Prediction of Polymeric Materials. *J. Phys. Chem. B* **2018**, *122* (1), 338–347. https://doi.org/10.1021/acs.jpcb.7b08461.

(4) OKUWAKI, K.; DOI, H.; MOCHIZUKI, Y. An Automated Framework to Evaluate Effective Interactionparameters for Dissipative Particle Dynamics Simulations Basedon the Fragment Molecular Orbital (FMO) Methodフラグメント分子軌道(FMO)法を用いた散逸粒子動力学シミュレーションのための有効相互作用パラメータ算出の自動化フレームワーク. *J. Comput. Chem. Japan* **2018**, *17* (2), 102–109. https://doi.org/10.2477/jccj.2017-0048.